

Examining Collaboration among Student Teams relying on Web Applications to Coordinate Software Development

Brian Boyd, Ayden Townsley, Charles Walter, Callen Johnson, Rose F. Gamble
Tandy School of Computer Science, University of Tulsa, Tulsa OK
[\[beb726, ayden-townsley, charlie-walter, callen-johnson, gamble\]@utulsa.edu](mailto:beb726, ayden-townsley, charlie-walter, callen-johnson, gamble}@utulsa.edu)

Abstract

Training students in software engineering should attempt to mimic industry practices. Thus, student teams develop non-trivial software products, which includes interacting with collaborative tools deployed as web applications. The interaction may be mechanistic or organic, and occur for different durations. Collaboration studies tightly control these factors, relying on manual activity logging, very specific software requirements, surveys and interviews. Since these tools allow simultaneous interaction and capture revision histories, collaboration may be more objectively measured. This paper investigates social media conversations, revision histories, and commit logs from undergraduate student teams performing software development. The objective is to examine how this form of data could be translated into collaborative activities and whether the same performance relationships are achieved in a class setting. A small pilot study shows that the translation methodology did not produce the exact relationships from other studies, but it does shed light on a team's perception of collaborators.

1. Introduction

Software development relies on team interaction to rapidly deploy high-quality, competitive products. Teams may be colocated or geographically distributed. Tasks may be well-defined or may have changing requirements. Project management may be hierarchical or ad hoc to allow for self-managed teams. Tools may be standardized and their use may be required, or teams may choose tools with which members have the most experience. All of these factors contribute how team members interact throughout the development process.

Several studies have shown that certain forms of collaboration lead to increased team productivity and team member satisfaction. For example, Robillard and Robillard [9] conclude that ad hoc collaborations, in

particular, increase the efficiency of collaboration. If industry studies can be controlled, they would allow for real-world data from experienced developers to be gathered and analyzed. But this is often very difficult to manage. Espinosa et al., [4] looked into the effect on asynchronous and non-collocated teams and found disadvantages to both in regards to collaboration, but also found that having a shared mental model could decrease the effect of both on collaboration. Collaborative platforms can have a positive effect on collaboration, especially if they can be easily learned [2].

What does collaboration look like for student software development teams? Such teams focus on the project part-time and engage in substantial asynchronous communication, as is commonly accepted among today's digital natives. Andres and Zmud [1] mention with regards to coordinating software projects, that students have the academic knowledge required to design and develop reasonably complex software within a practical amount of time. Earlier studies help to elaborate on how student-controlled studies typically unfold. These studies are often scaled-down to accommodate the limits of students compared to industrial software development teams. However, in specific regards to collaboration, student-controlled studies are invaluable to gathering authentic, raw data. Collaboration between students is expected to be high because teams are made up of peers, so it is assumed that collaborative interactions would be evident and easy to quantify.

In this paper, we investigate how team performance is affected by collaborative activities given particular measures. The objective is to single out factors that may positively or negatively affect performance, such as the team grade and the quality and creativity of the developed software. By understanding what constitutes collaboration and its effects, instructors can find ways to directly encourage collaborative work so that its benefits are realized throughout software engineering education and training.

Our approach examines captured interactions from multiple collaborative tools: GitLab, Google Drive, Lucidchart, and Slack.com. These tools are used in a

software engineering class in which principles and practices, along with tool usage, are taught within a fall semester and followed by dedicated team projects within the subsequent spring semester. We define what constitutes evidence of collaboration within the interactions by defining analogies to metrics of collaboration previously identified to ensure consistent measurements and data analysis [1, 2, 4, 9]. The collected data is aggregated from the conversations, revision histories, and commit logs of the collaborative tools to produce evidence of collaboration. We correlate the evidence with metrics that have previously indicated higher performance levels within the same class structure to answer the following research question.

RQ: *Do objective collaboration measures taken from log files of collaborative tools used as a natural part of a class correlate to team performance with respect to project grades, engagement in the project, and peer evaluation?*

The remainder of the paper is organized as follows. Section 2 presents background on collaboration studies from both industry and academia. Section 3 overviews the broader research goals and presents the experimental setting. Section 4 discusses the team performance metrics, the translation of collaboration metrics from the background studies to the tool log files, and the analysis performed. Section 5 presents the outcome of the evaluation. Section 6 provides further discussion and concludes the paper.

2. Background

Various types of collaborative efforts have unique associations with the work done by a team. We rely on the study by Robillard and Robillard [9] and their definitions of *individual*, *ad hoc*, *called*, and *mandatory* as the types of collaboration that can be observed and measured. Individual work is performed independent of other team members. Ad hoc collaboration is spontaneous work done with at least one other person in the room. Ad hoc collaboration, in particular, has a significant effect on the communication of the team. Called collaboration is a planned, but not required, working session involving one or more members of the team. Mandatory collaboration is established through required meetings usually involving the entire team. The study by Robillard and Robillard [9] used logbooks kept by a team as required for the project. The defined collaborative types were extracted from the data. To help define the types of collaborations, they used time,

ID, duration, number of team members, phase, activity, task, and comments as metrics to separate the data. Ad hoc collaboration accounted for the largest share of time spent collaborating, followed by individual work. Mandatory and called collaboration contributed little to the overall communication of the team. We will return to these collaboration types and their definition with respect to the data that we capture to understand similar aspects of the effectiveness of collaborative efforts in software development.

In defining effective collaboration, especially as it relates to military product development, Noble, et al., [8] discuss four specific categories of collaboration based on the efficiency of the team in delivering a product. These categories (1) address product quality and team efficiency, (2) team behaviors, (3) group understandings, and (4) individual team member understandings. They define collaboration as “the methods of people actively sharing data, information, knowledge, perceptions, or concepts when working together towards a common purpose.” The existence of specific collaboration metrics could provide a reliable way to measure how effectively the team collaborates and if it yields a collaborative product. The two types of collaboration they focus on are cognitive and non-cognitive collaboration. Cognitive collaboration deals with behavioral and team focused data, whereas non-cognitive collaboration uses the resulting product as the measure of the collaboration. Though cognitive collaboration cannot be discounted, non-cognitive collaboration should be easier to objectively measure, possibly through automated means. For our pilot study, we relied on non-cognitive collaboration behavior as represented by team member usage of interactive services while they developed their overall product. The objective was to determine if such measures could be achieved and investigated in the natural project setting of an undergraduate team, without introducing additional behavioral measurement tools.

Bjorn and Ngwenyama [2] examined how to efficiently manage teams and their communications, using traditionally non-collaborative tools. They conclude that email and other forms of communication do not allow for a “collaborative configuration” and therefore hinder the translucence of the overall communications. To collect their data, they focused on interviews with the participants, concentrating only on personal experience without looking at an objective metric. They emphasized “groupware technology” as a primary tool to increase communication through a virtual medium.

Andres and Zmud [1] concluded that organic coordination leads to more successful projects than

mechanistic coordination. Eighty subjects, drawn from a population of undergraduate students, were used to conduct their study. The students were required to develop two software subsystems, where the duration for each task did not exceed six hours. The authors looked at whether the information processing needs of the work group fit the information processing capacity associated with the coordination strategies that were used. The results showed that informal, cooperative, and decentralized coordination strategies can be considered organic coordination, while formal, controlling, and centralized coordination strategies can be considered mechanistic coordination.

Espinosa et al., [4] examined if shared mental models have a positive correlation to efficient collaboration. They conducted surveys, coupled with measures of time and familiarity, to conclude that shared mental models should lead to increased positive collaborative efforts. They looked at both collocated and non-collocated teams and found that, when the added time between material updates from non-collocated teams was taken into account, the teams were still able to maintain a shared mental model, leading to improved collaboration. Despite our teams being collocated, much of the communication is done via services used by non-collocated teams. This result shows that they are still able to maintain good collaboration regardless of the physical location of the team members.

Zeiller and Schauer [10] examined the factors that lead to success with incorporating social media into small to medium-sized enterprises. They defined collaboration as the joint work of a group of people on shared objects, where the users share a common goal and are jointly responsible for the outcome. They relied on interviews with key users involved in the project for data collection. They found that the two most important factors for social media utilization success were management support and support for users.

He et al. [6] attempted to put empirical data behind the theory of team cognition. They monitored several teams while they completed a task, focusing on communication for collaboration. The tasks were designed such that the only measurable result was collaboration. They were unable to fully nail down an empirical measurement of the concept.

Coman et al. [3] defined cooperation as help being provided between teammates sharing the same goal towards solving an issue. They looked at two independent field studies, comparing solo programming to pair or team programming. They found that around 40% of the time developers will work in pairs or help each other. This help was a form of ad hoc communication, as it was not planned prior to the need

for help arising. An additional result was that the use of informal pair-programming complemented solo coding efforts.

Using Jazz, a proprietary research platform created by IBM which embeds collaborative capabilities into the Eclipse development environment, Hupfer et al. [7] investigated potential advantages to using the collaborative system as opposed to the traditional non-collaborative version. They focused on contextual collaboration, which they define as an approach to collaboration in which users are not forced to leave their core applications to launch collaborative tools. Our study uses Slack as a central hub for collaborating between users, which had useful unobtrusive collaboration through features such as GitLab integration.

3. Experimental Environment and Methodology

If we can ascertain collaboration metrics from data that objectively details team interaction and these metrics can alert to potential issues in productivity and product quality, then teams and management can address the issues early in the software development process. For a class setting, it can be an important factor in the success of the project and the learning process of the team. However, it is not clear that the results from earlier collaboration studies can be translated to team interaction within a class. These earlier studies are survey-based, have manual logging of activities, or highly controlled requirements for the software that is developed with short time-frames. Actual team engagement across weeks of development time may manifest different forms of activities that could be considered collaborating.

In addition to evaluating the research question in Section 1, the goals of the pilot study we report on in this paper are as follows.

1. Determine if we had sufficient evidence of collaborative activities according to the previously defined collaboration types by Robillard and Robillard [9] within the Slack.com conversation history.
2. Construct scripts to automate the data conditioning to represent duration and percentage of team members collaborating at specific times using Google Drive and LucidChart revision histories and GitLab commit logs.

3. Investigate what it means to collaborate through task-related, possibly asynchronous, conversations on social media.
4. Determine if the individual performance metrics found in [5] can be reused from a team perspective and what correlations to these metrics means to the team.

3.1. Class Setting

To understand collaboration and its effects in a software engineering class that uses multiple online and interactive tools, we investigate the relationship between metrics associated with collaboration and performance metrics used within the class.

The undergraduate class targeted is a capstone course required of all computer science, information technology, and simulation and gaming majors. Teams develop a non-trivial software product for an external customer, which is often a local or regional non-profit organization. The products are mobile or web applications.

The spring semester dedicated to the team projects is partitioned into three sprints that follow a modified Agile Scrum development process. There are two major modifications to the process. One modification is that with the exception of 3-4 meetings per semester when the customer is present, the instructor serves as the customer's representative on the team. The second modification is the sprint daily meeting, which contributes to the participation metric discussed in Section 4.1. Because the students are taking other classes and, as seniors, are focused on graduating and life beyond graduation, a daily meeting is overly burdensome. The teams are therefore required to log their meeting check-ins in their team's Slack.com account at designated periods during the week, which are usually two times per week. There are mandatory face-to-face meetings throughout the sprints, which are documented within the online meeting check-ins.

We examine each sprint as a separate entity because they are mostly self-contained. The deliverable for a sprint focuses on satisfying particular user stories and tasks, so the work products required for each sprint only minimally overlap. For the pilot study, we used data collected from 5 teams in the same software engineering class, yielding a total of 15 evaluation points. Teams had either 3 or 4 team members and data was normalized to reflect team size. Evaluations of the individual team members are required to be performed by each team member after each sprint.

To extract collaboration-related metrics from team activities, we examined content and events from team interactions with Slack.com, Google Drive, Lucidchart, and GitLab, as outlined below. Each of these contribute to work products within each sprint. An event consists of a post to Slack.com. An event in GoogleDrive and Lucidchart is any revision that is saved with a timestamp. In GitLab, an event is a code commit. Overall, 12,574 events were used as part of the study.

Slack.com. In addition to being an internet relay channel, Slack.com allows documents to be directly shared as part of the conversation. Users can communicate on public or private channels and also directly with other users. New channels can be created by users to differentiate among conversation threads. In the class, each team had their own account, administered by the instructor. Teams were expected to communicate via public channels because private and direct communications cannot be exported. Posts in public channels were exported as JSON, placed in a spreadsheet, and evaluated by three individual raters to determine when statement content reflected collaborative activity in the form of Mandatory, Called, Ad hoc, or non-collaborative (Individual) given the definitions provided in Robillard and Robillard [9]. To maintain consistency, the three raters independently performed a manual classification of the Slack.com posts based on an agreed upon set of examples representative of each collaborative type. The group met to finalize any disagreements in the ratings, which were less than 5% of the total number of posts.

Google Drive. Google Drive was used for product and sprint backlogs (Google Sheet); general documents for executive summary, risks, user guides, and troubleshooting (Google Doc); and presentations to the class and customers (Google Slide). It is ideal for collaborative document changes because it allows for changes by multiple users at the same time. Each Google drive work product was owned by the instructor and shared with the teams to allow for capturing the revision history, which was studied as part of defining collaboration metrics.

Lucidchart. Models and wireframes were constructed using Lucidchart, which is a web-based tool that has multiple templates, including UML diagrams. Like Google Drive, it allows multiple users to edit the model pages at the same time and maintains a revision history. Each Lucidchart work product was owned by the instructor and shared with the teams to preserve and capture the revision history which was analyzed for collaborative activity.

GitLab. Version control is an important part of software development. We deploy GitLab on an in-

house server that each team member must commit to from their personal account. GitLab logs the commits and associated messages to allow for analysis.

4. Metrics and Analysis

In this section, we overview individual performance metrics that have been successfully used in the class for a number of years. We provide a team perspective of the same metrics to answer the research question in Section 1. Using the definition of collaboration metrics from Section 2, we translate the measurable factors into representations of collaboration given social media posts, work product revision histories, and commit logs. We then review the correlation analysis providing the raw data from a small pilot study.

4.1. Team Performance Metrics

In a software engineering class, **grades** are the primary measure of the team's performance with respect to meeting the requirements of the Scrum sprint as outlined in the detailed rubrics. Sprints range from 500-900 available points. The large number of points allows a team potentially fail one aspect of a sprint, yet obtain and acceptable grade by doing exceptional work on another aspect of the sprint. Grade calculations are based on the quality of the work products and the satisfaction of the rubrics. The rubrics include but are not limited to the task breakdown and responsibility allocation in the Sprint backlog, user story and use case specifications, models and wireframes, coding practices, user interface, and team presentation.

Creativity and **quality** scores are determined using a 10-point scale as answers to a set of questions posed

to the class (i.e. peer review) and the team's customer at the end of the sprint regarding the deliverable functionality and the user interface. The product can be examined and tested based on scripts provided by the team that focus on the actual functionality being delivered at a given time. The instructor and teaching assistants, who are graduate students, also rate creativity and quality using the same questions. Creativity has proven to not only be hard to define when viewing a product presentation and demonstration, but also hard to assess. Similarly, quality is often evaluated by the user interface or user experience with the product during testing and is unrelated to the actual programming of the functionality. Both criteria scores are normally integrated into the sprint grade, but we separate them out here in an effort to determine if there is a relationship with collaboration.

Participation, Event Proportion, and Impression have been studied in terms of individual performance metrics using similar posting and log files [5]. In this paper, we redefine them in terms of a team perspective in order to put them at the same level of assessment as the collaboration metrics and the team grade, creativity, and quality assessments.

Average Participation. Participation is scored according to the content, requirements fulfillment, and timeliness of a person's meeting check-in on Slack.com. Individual participation in meeting check-ins has been directly associated with a person's overall performance, as well as how they are characterized by their peers with respect to skills and work ethic. Teams with members that have high individual participation scores have increased project success rates and better team satisfaction. To provide a team perspective of participation, we averaged the participation score percentages of all team members.

Table 1. Raw data collected from collaborative tools per team

	Sprint 1					Sprint 2					Sprint 3				
	Team 1	Team 2	Team 3	Team 4	Team 5	Team 1	Team 2	Team 3	Team 4	Team 5	Team 1	Team 2	Team 3	Team 4	Team 5
Grade	0.89	0.77	0.69	0.81	0.89	0.98	0.99	0.86	0.87	0.94	0.88	0.99	0.90	0.98	1.00
Creativity	0.75	0.45	0.53	0.50	0.77	0.85	0.98	0.58	0.67	0.60	0.56	0.84	0.72	0.84	0.90
Quality	0.82	35.00	40.00	45.00	40.00	1.00	1.00	0.80	0.75	1.00	0.89	0.84	0.81	0.71	0.93
Avg. Participation	0.63	0.81	0.81	0.91	0.93	0.94	0.94	0.63	0.96	0.97	1.00	1.00	0.75	0.75	1.00
Avg. Events	544.75	168.75	92.50	136.00	168.00	618.75	264.00	117.75	106.25	221.33	314.75	142.25	95.25	177.33	100.75
Avg. Impression	0.81	0.96	0.81	0.86	0.95	0.90	0.97	0.79	0.67	0.97	0.94	0.98	0.78	0.66	0.96
Mandatory collaboration posts	0.03	0.13	0.32	0.35	0.44	0.02	0.14	0.15	0.18	0.27	0.05	0.10	0.07	0.08	0.33
Called collaboration posts	0.06	0.05	0.12	0.03	0.14	0.08	0.10	0.07	0.02	0.14	0.07	0.22	0.01	0.06	0.10
Ad hoc collaboration posts	0.40	0.28	0.07	0.08	0.01	0.45	0.05	0.13	0.18	0.18	0.44	0.04	0.26	0.62	0.15
Non-collaborative posts	0.52	0.54	0.48	0.54	0.41	0.44	0.70	0.65	0.62	0.41	0.44	0.64	0.66	0.24	0.42
Instances of collaboration	14.00	11.00	6.00	8.00	13.00	39.00	28.00	12.00	8.00	10.00	29.00	7.00	4.00	5.00	7.00
Average duration of collaborations	18.18	7.54	13.47	7.53	10.32	22.70	18.67	10.97	7.38	17.20	23.72	12.63	5.00	6.60	10.79

Average Events. Event proportion for an individual is the number of overall events or logged activities performed by a person with respect to the total number of events performed by the team. Higher event proportion scores have been directly related to better overall performance and better product development within the class. Individuals that have higher event proportion scores are also more revered by their peers. To reflect this as a team score, we took the total number events and divided it by the number of team members.

Average Impression. An impression score is given to an individual based on self and team evaluations. Peer evaluations are conducted after each sprint in which individuals rate themselves and each team member on a set of qualities. For each quality Q, each individual is assigned a per-sprint score equal to the average score that the individual achieved on Q over all the evaluations received by team members. Example questions on the self-evaluations appear below.

- Q1: Participated in online and face-to-face meetings
- Q2: Responded to team activity in a timely manner
- Q3: Contributed project ideas that were implemented
- Q4: Contributed to document artifacts creation and/or review
- Q5: Set the standard for team performance
- Q6: Completed all tasks assigned at agreed upon timeline
- Q7: Organized the sprint artifacts, user stories, & sprint backlog
- Q8: Communicated project and instructor expectations well
- Q9: Performed design based tasks effectively
- Q10: Initiated and maintained required documents
- Q11: Worked with team to create overall product vision
- Q12: Delegated tasks appropriately

To provide a team impression score, we took the average impression score percentage of the team members.

4.2. Acquiring Collaboration Evidence

To examine the evidence, we quantified the following aspects of collaboration analogous to metrics found in Section 2 given our data set. Below we use the previous metric terminology and our analogous data collection methodology.

Each post in Slack.com was partitioned into a collaboration type: Mandatory, Called, Ad hoc or Non-collaborative (Individual performance) as defined by

Robillard and Robillard [9]. We then normalized the post count by calculating the percentage of that post type given all of the posts. The normalization to percentage was necessary because of the different team sizes and to avoid bias toward teams with more members who should naturally have more posts. The individual types are detailed below.

Mandatory (Mechanistic) collaboration. Because meeting check-in was a required communication on Slack.com for each team member before a predefined deadline, check-ins served as mandatory collaboration. Not all team members perform the check-in, despite the requirement. If team members have Slack.com notify them when posting occurs, often once one member checks in, the rest will follow. This leads to further conversation regarding everyone's progress on the project. Since these posts are closely related to the check-in, they retain mandatory collaboration status.

Called collaboration. This type of collaboration would be indicated on Slack.com when a team member schedules a meeting and gets some portion of the team to agree to a time, place, and general agenda or reason for meeting. These indicate that additional untracked collaboration occurred between the team members involved, as the collaboration can happen in person or using Google Drive, Lucidchart, or GitLab.

Ad hoc (Organic) collaboration. When any subset of the team members with at least two members report on Slack.com that they met to work on a particular product feature or work product, discuss the need to meet, or noted that they talked with the instructor or TA about an issue, these events are counted as ad hoc collaboration. For example, posts that include "I'm in the lab now if anyone's around and wants to work on the wireframe," followed by another team member posting "I'm on my way there in 10 minutes" are considered ad hoc collaboration unless they are followed by a post indicating that no one could make it.

Non-collaborative (Individual work) posts. These posts exhibited no collaborative activity. They were generally team members discussing individual progress or problems with the project, as well as discussing life and other classes.

Instances and duration of collaboration. To investigate the instances and duration of collaboration, we examined the revision histories of Google Drive and Lucidchart, along with the commit log for GitLab. A collaboration event recording starts when two or more users are present in the history or log file within at most 30 minutes of each other. This interval was determined by calculating the average number of total events across all teams per sprint and choosing a team/sprint closest to that average. For this team and sprint, we calculated

the time between collaborative events using manual observation as 23.72 minutes. We assumed that a 30-minute interval was acceptable to catch when a second person joined with the first to engage in a collaborative activity. The duration of a collaboration event was assumed to have ended if greater than 30 minutes passed before the next revision or commit. We retained the 30-minute interval assuming the collaborative activity would have continued given our earlier average of how long it would be before a second team member became active.

We summed the collaborative instances across Google Drive, Lucidchart, and GitLab work products and left them in their raw form. Thus, the total number of instances of collaboration is the sum of the total number of recorded collaborations that occur within a sprint. Because the study is small, we show the raw data in Table 1.

4.3. Analysis

If collaboration is defined analogously to prior studies, then it should be the case that the correlations are retained at some level. We examine correlations using Pearson's correlation coefficient r , computed using sum of squares. For a two-tailed test with 13 degrees of freedom, given 15 evaluation points, the following critical values were computed:

- $|r| > 0.441 \Rightarrow p < 0.1$
- $|r| > 0.511 \Rightarrow p < 0.05$
- $|r| > 0.641 \Rightarrow p < 0.01$

5. Outcome of Evaluation

The result of the correlation is shown in Table 2. Dark lines surround the cells with significant correlations, which are

- a positive correlation between mandatory collaboration posts and quality
- a positive correlation between called collaboration and average impression
- a positive correlation between average duration of collaboration and average number of events
- a positive correlation between ad hoc and overall instances of collaboration
- a negative correlation between mandatory collaboration and average number of events

The first correlation to note is between mandatory collaboration posts and quality. We expect the relationship may be partially due to the requirement that the team members must discuss specific areas where they are having problems and what they will get done before the next post in mandatory posts. This requirement is intended to encourage other team members to provide direction, advice, or help, possibly leading to a cooperative effort that produces higher product quality.

The correlation of called collaboration and average impression is interesting, because impression generally dictates the amount of confidence and trust team members have for each other's capabilities. Called collaboration is additional scheduling of team activities. A relationship between these two metrics may imply that collaboration is increased (decreased) with respect to the positive (negative) feeling team members have for each other.

The average duration of collaboration being correlated with the number of events shows that, when users collaborate for extended periods of time, they must continue to communicate. This high correlation is expected for teams that communicate frequently. The correlation can also be tightly coupled with the event count itself, since collaboration events are part of that event count.

The correlation between ad hoc and overall instances of collaboration with the average number of events is also in line with general expectations. For ad hoc collaboration to be successful, there is a need for a large amount of discussion between group members to arrange the meetings. This is similar with the instances of collaboration. The more collaboration events that the teams have, the more events they are likely to have overall. These were used to verify that our metrics were working as expected, taking into account the size of the study.

Somewhat unexpectedly, the negative correlation of mandatory collaboration and average number of events suggests that if a team communicates at set intervals and with predefined goals, they are much less likely to continue discussion or communicate in other ways. This leads to a lower number of overall events, potentially lowering other benefits of the team collaboration. There is a need to strike a balance between ensuring that a project is on track through mandatory collaboration and allowing the team to reap the benefits of other forms of collaboration.

Table 2. Correlation of collaboration metrics with team performance metrics

	<i>Grade</i>	<i>Creativity</i>	<i>Quality</i>	<i>Avg. Participation</i>	<i>Avg. Events</i>	<i>Avg. Impression</i>
<i>Mandatory collaboration posts</i>	-0.297	-0.186	0.656	0.305	-0.532	0.214
<i>Called collaboration posts</i>	0.281	0.315	-0.001	0.389	-0.015	0.593
<i>Ad hoc collaboration posts</i>	0.210	0.056	-0.414	-0.273	0.526	-0.388
<i>Non-collaborative posts</i>	-0.114	-0.024	-0.099	-0.102	-0.188	0.071
<i>Instances of collaboration</i>	0.283	0.252	-0.238	0.264	0.756	0.364
<i>Average duration of collaborations</i>	0.249	0.184	-0.323	0.266	0.745	0.455

6. Discussion and Conclusion

With respect to the originally posed research question, the existence between called collaboration and average impression suggests that there exists a relationship between collaboration and peer evaluation. Given that it is called collaboration, it may indicate that those team members who schedule interactions may be viewed more positively. The other positive correlations suggest a relationship with project engagement and indicate that collaborative teams do more work, as seen in a correlation with average number of events, which may lead to additional instances of collaboration.

The lack of correlation of any of the collaboration metrics with the team grade is notable. Given that the computer science major is accredited, there are very specific rubrics that must be satisfied within the course and these are detailed in each of the sprints. Upon reviewing the results, it was evident that the rubrics are focused on meeting criteria such as deliverable functionality and properly configuring the project and sprint backlogs. The study results show that more attention is needed to foster collaboration, perhaps through some form of grade reward.

In the study by Robillard and Robillard [9], the highest level of collaborative activity across their teams was ad hoc, followed by individual work. If we average the percentages of each collaborative post type across all teams in Table 1, the highest level of activity is individual work, followed by ad hoc, with mandatory a close third. Similar to [9], called collaboration adds little to the collaborative effort.

Besides the small number of evaluation points, there are some other potential issues with the data collection that could have exposed problems in the correlation. One issue is that we examined the types of collaboration only in Slack.com conversations and the duration of collaboration only in Google Drive, Lucidchart, and GitLab. We did not use both types of collaboration metrics on all data collected. In addition, there was no cross referencing. It is possible that a collaboration that

started within one work product continued into another work product. It is also possible that collaborative discussions in Slack.com may have continued to work products and vice versa but not be mentioned.

Another issue is the averaging of individual metrics in Section 4.1 to form a team score. When a “social loafer” is on the team or one team member is overly busy or has less skills, their lower individual performance scores are masked by the other team member scores. This may affect the correlations identifying other relevant relationships. A larger study is underway, bringing in past years of data to see if the correlations change.

Automating the post filtering to more easily obtain the counts of different collaboration types is part of the future challenges of the work. It is possible to filter out posts that use the phrase “check-in” referring to a mandatory check-in. We can also filter out where another person is explicitly mentioned in a post, along with posts that contain “meet”, “let’s”, “team”, pronouns, and words referring to time. However, the biggest type of automated discernment difficulty is between called and ad hoc collaboration. One example of this difficulty is seen in the following post. “Hey guys, I’m done with the xml though I need @User2 to come by the lab while I’m there to check the duration and delay time before I commit and push.” This post would be tagged because it has another team member’s name. But there are no additional key words for it to be considered called collaboration because it has no time to meet, which does not appear until the conversation progresses.

Face-to-face meetings are not captured using our approach. As future work, we will be investigating the use of web cams during face-to-face meetings so that they can be reviewed to determine the collaboration event types embedded in such meetings. The use of Google Hangouts, which can record a video of a meeting, can also be examined for collaboration events when the teams are not in the same location but prefer to have “face time”.

A potential outcome of this form of research is that if positive collaboration can be objectively measured as part of the tools, then algorithms can be developed to automate the collaboration metrics analyses. These algorithms could eventually become part of the functionality of the tools. For example, both GitLab and Slack.com produce their own form of usage statistics.

7. References

- [1] H. P. Andres and R. W. Zmud, "A Contingency Approach to Software Project Coordination", *Journal of Management Information Systems*, 2002, pp. 41-70.
- [2] P. Bjorn and O. Ngwenyama, "Virtual team collaboration: building shared meaning, resolving breakdowns and creating translucence", *Information Systems Journal*, 2009, pp. 227-253.
- [3] I. Coman, P. Robillard, A. Sillitti, and G. Succi, "Cooperation, collaboration and pair-programming: Field studies on backup behavior", *Journal of System Software*, Elsevier Science Inc., 2014, pp.124-134.
- [4] A. Espinosa, R. Kraut, S. Slaughter, J. Lerch, J. Herbsleb, and A. Mockus, "Shared Mental Models, Familiarity, and coordination: A Multi-Method Study of Distributed Software Teams", *ICIS 2002 Proceedings*, 2002, pp. 425-433.
- [5] R. Gamble and M. Hale, "Assessing Individual Performance in Agile Teams", *Proceedings of the 43rd Annual Frontiers in Education Conference*, 2013, pp. 1678-1684.
- [6] J. He, B. Butler, and W. King, "Team Cognition: Development and Evolution in Software Project Teams", *Journal of Management Information Systems*, M.E. Sharpe, Inc., 2007, pp. 261-292.
- [7] S. Hupfer, L. Cheng, S. Ross, and J. Patterson, "Introducing Collaboration into an Application Development Environment", *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work*, ACM, Chicago, Illinois, USA, 2004, pp. 21-24.
- [8] D. Noble, D. Buck, J. Yeargain, "Metrics for evaluation of cognitive-based collaboration tools", *International Command and Control Research and Technology Symposium*, 2000.
- [9] P.N. Robillard and M.P. Robillard, "Types of collaborative work in software engineering", *The Journal of Systems and Software*, vol. 53, 2000, pp. 291-224.
- [10] M. Zeiller and B. Schauer, "Adoption, motivation and Success Factors of Social Media for Team Collaboration in SMEs", *Proceedings of the 11th international conference on Knowledge Management and Knowledge Technologies*, ACM, Graz, Austria, 2011, pp. 4:1-4:8.